Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики
Базовая кафедра вычислительных и информационных технологий

УТВЕРЖДАЮ
Заведующий кафедрой
_____ /В.В. Шайдуров
«21» июня 2016 г.

БАКАЛАВРСКАЯ РАБОТА

Направление 02.03.01 Математика и компьютерные науки

СИСТЕМА ВИЗУАЛИЗАЦИИ ИСТОЧНИКОВ ИНТЕРНЕТ-ТРАФИКА

Научный руководитель
Кандидат технических наук,
доцент _____ / С.В. Исаев
16.06.2016

Выпускник _____ / Г.К. Сурихин
16.06.2016

Красноярск 2016

РЕФЕРАТ

Выпускная квалификационная работа по теме «Система визуализации источников интернет-трафика» содержит 31 страниц текстового документа, 7 рисунков, 1 таблицу, 19 использованных источников литературы.

ИНТЕРНЕТ-ТРАФИК, WHOIS-СЕРВЕР, TRACEROUTE, IP-АДРЕС, КАРТОДИАГРАММА, ВИЗУАЛИЗАЦИЯ, БАЗА ДАННЫХ.

Цель работы – построить систему способную визуализировать информацию об источниках интернет-трафика, с применением географической карты.

В работе исследованы способы получения информации относительно IP-адресов. Разработан соответствующий алгоритм, для поиска информации про список IP-адресов с большим количеством записей. На основе этого алгоритма создано клиент-серверное приложение, с использованием наиболее современных технологий, способное визуализировать информацию об источниках интернет-трафика.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Каждый пользователь, в глобальной сети интернет, идентифицируется уникальным номером, называемым IP-адресом. В свою очередь, IP-адрес может быть соотнесен с информацией о пользователе, такой как адрес, телефон и другие контактные данные. Стоит отметить, что в некоторых случаях, информация может отсутствовать, по причине делегирования того или иного адреса каким-либо структурам, которые предпочитают не раскрывать информацию о себе. Существует большое количество задач, в которых нужно получить информацию о большом количестве адресов и по возможности визуализировать ее. Наиболее актуален этот вопрос в сфере, где работа идет с большим количеством пользователей (веб-сервисы, сайты) и часто встает вопрос о выявлении недоброжелательной активности к вашему ресурсу.

Наиболее успешным сервисом, в данной отрасли, является Google Analytics, однако, он больше направлен на предоставление информации для владельцев сайтов, с целью анализа пользовательского сегмента и его последующего увеличения. Также, отсутствует возможность отобразить данные на карте, что на мой взгляд, является важным условием к понимаю ситуации в целом.

Цель бакалаврской работы – создание приложения, способного по IP-адресам, полученным из журнала работы сервера, построить картину их местоположения. Реализовать возможность фильтрации адресов, по таким критериям как регион, количество и время запросов. Также, в данном приложении предполагается возможность получения более полной информации, касательно конкретно IP-адреса, по запросу пользователя.

Система состоит из 2-х главных модулей. Первый – это сервер (реализованный с помощью средств node.js) , отвечающий за поиск информации для заданного набора адресов. Второй – это визуальная часть

приложения (реализована с помощью средств AngularJS), визуализирующая информацию, присланную с сервера.

1. Интернет-трафик и его роль

1.1 Основная проблематика

Подключив свой компьютер, к глобальной сети интернет, вы, вместе со всеми теми возможностями, что он предоставляет, делаете себя более уязвимыми для внедрения злоумышленников, желающих украсть вашу личную информацию или преследующих иные корыстные цели. Особенно это важно, в случае постоянного обмена, большим количеством данных с сетью интернет. Часто это бывает, когда в роли пользователя выступает организация, по долгу службы, обязанная работать с большим числом абонентов, в частности принимать от них запросы и отправлять соответствующие ответы (сайты, почтовые сервера и т.п.). В этом случае, важно бывает, заранее предусмотреть, по каким-либо признакам или странной активности в сети, возможные атаки или недобросовестные действия, так как под угрозой может оказаться не только информация какого-либо одного конкретного пользователя, а целой группы. Основным источником для анализа выступает сетевой трафик. Здесь стоить оговориться, что в данной работе, речь идет не об анализе сетевого трафика как такового, а об анализе его источников и маршрутов следования. Помимо прочего, анализ сетевого трафика может помочь, не только с точки зрения предотвращения атак злоумышленников, но и в коммерческих целях. Примером этого, может служить анализ посетительской активности на сайте, с целью выявления предпочтений или целевой аудитории по каким-либо критериям. Встает вопрос – как правильно анализировать сетевой трафиков, какие технологии при этом использовать и как подавать информацию в финале?

Одним из способов решения, вышеупомянутой задачи, а именно – удобное представление данных об источниках интернет-трафика, является его территориальная визуализация на географической карте.

Наглядное представление источников, дает понимание картины в целом и предоставляет возможность для дальнейшего прогнозирования. Среди наиболее известных утилит, занимающихся анализом трафика и с ним смежных данных, в процессе работы, рассмотрены следующие:

- Wireshark – анализатор сетевых протоколов сетей, с возможностью захвата трафика в режиме реального времени и удобным пользовательским интерфейсом [1];
- ClearSight Analyzer – захват большого количества трафика с возможностью индексации и в дополнение имеет множество настроек для анализа, такие например как установление различных пороговых значений для отклика сервера [2];
- Google Analytics – является веб-утилитой, больше направлена на анализ действий пользователей на сайте, но также предоставляет возможность анализа некоторых сегментов трафика [3].

Из перечисленных утилит, ни одна, не предоставляет возможность визуального географического вывода, с использованием карт или иных топографических объектов. Под визуальным представлением, здесь понимается, вывод данных на карту, с сортировкой по регионам источников трафика.

1.2 Визуализация

Средствами изображения территориального размещения являются штриховка, фоновая раскраска или геометрические фигуры. Различают картограммы и картодиаграммы.

Картограммы - это схематическая географическая карта, на которой штриховкой различной густоты, точками или окраской определенной степени насыщенности показывается сравнительная интенсивность какого-либо показателя в пределах каждой единицы нанесенного на карту территориального деления (например, плотность населения по областям или республикам распределения районов по урожайности зерновых культур и т. п.). Существуют фоновые и точечные картограммы.

Вторая большая группа – это картодиаграммы, представляющие собой сочетание диаграмм с географической картой. В качестве изобразительных знаков в картодиаграммах используются диаграммные фигуры (столбики, квадраты, круги, фигуры, полосы), которые размещаются на контуре географической карты. Картодиаграммы дают возможность географически отразить более сложные статистико-географические построения, чем картограммы[4].

Стоит отметить, что нам придется работать с размещением большого количества объектов на карте, т.е. возможно наложение и загромождение объектов.

В картографии наиболее употребительны: линейные диаграммы - столбики, полоски и т. п., длина которых пропорциональна сравниваемым величинам; площадные диаграммы - квадраты, круги и т. п., площадь которых пропорциональна сравниваемым величинам; объемные диаграммы - кубы, шары и т. п., объем которых пропорционален сравниваемым величинам. В то же время диаграммные фигуры могут быть структурными, если, например,

квадраты, круги и другие фигуры подразделяются на части (аналогично суммарным значкам) соответственно составу (структуре) изображаемого явления.

О соотношениях величин лучше всего судить по линейной диаграмме, но она не экономна по размерам фигур. Меньше места требуют площадные и особенно объемные диаграммы: различия между их наибольшими и наименьшими фигурами не так значительны. Объемные диаграммы удобны для сильно различающихся величин. Однако зрительно соизмеримость площадных и тем более объемных диаграмм не так очевидна - различия в площадях и объемах представляются на картодиаграмме меньшими, чем в действительности[5].

Сопоставление величин облегчается, когда они показаны группами равнозначных фигурок (кружков, квадратиков, прямоугольников и т. п.), из которых каждая обозначает определенное количество единиц изображаемого явления.

Учитывая вышесказанное, самым оптимальным решением вопроса визуализации источников, является использование площадных структурных диаграмм.

Теперь необходимо, углубиться и конкретизировать объект, с которым нам предстоит работать, а именно сетевой трафик и какую информацию про него мы можем получить. Данные в сети всегда идут из одного места в другое, т.е. существует начальная и конечная точки, следовательно, существует адрес, по которому эти данные идут. Этот адрес в сети интернет называется ip-адресом. Соответственно наша задача, заключается в том, чтобы IP-адрес сопоставить реальному адресу, и отобразить на карте.

2 Адресное пространство в глобальной сети интернет

2.1 IP-адрес

Для начала, нам необходимо понять объект, с которыми предстоит работать, а именно IP-адресом.

IP-адрес является уникальным идентификатором (адресом) устройства (в основном, компьютера), которое подключено к глобальной или локальной сети. IP-адреса представлены 32-битовыми (версия IPv4) или 128-битовыми (версия IPv6) двоичными числами. Удобная форма записи IP-адреса (версии IPv4) – это запись в виде 4-х десятичных (от 0 до 255) чисел, которые разделяются точками [6], с последними нам и предстоит работать. Стоит оговориться, что адрес устройства в сети может быть представлен как последовательность букв (пример: www.google.com), так называемый домен, введен с целью упрощения запоминания. За предоставление соответствия буквенного и числового представления отвечает DNS (Domain Name System) [11].

Вопросы получения того, или иного адреса, решается интернет-провайдером (регистратором) данного региона, который, в свою очередь, получает пул доступных для него адресов от своего провайдера.

Вертикаль делегирования адресного пространства, в сети интернет, устроена следующим образом:

Поверх всего стоит организация под названием IANA (Internet Assigned Numbers Authority) . Она в свою очередь делегирует часть адресного пространства региональным интернет-провайдерам (всего их 5), а они далее по цепочке более мелким провайдерам и т.д. В общем и целом, IANA занимается тем, что решает вопросы выдачи доменных имен, управления данных корневых серверов DNS и смежные с этим задачи[7]. При выделении адресов запрашивается информация, записанная в базе данных.

2.2 Whois-серверы

Остается открытым вопрос, как узнать необходимую нам информацию относительно IP-адреса. Для этих целей существует whois-сервер. Whois – сетевой протокол, базирующийся на протоколе TCP. Его основное предназначение – получение в текстовом виде регистрационных данных о владельцах IP-адресов и доменных имен. Записи об IP-адресах сгруппированы по диапазонам (например, 8.8.8.0 — 8.8.8.255) и содержат данные об организации, которой этот диапазон делегирован[12]. Для начала работы с ним, необходимо открыть TCP соединение на порт 43 к нужному whois-серверу. Далее послать запрос в определенном формате (который для конкретного whois-сервера может быть каким угодно), закончить его "\r\n" и получите результат, формат которого для конкретного whois-сервера также может быть каким угодно. Закрытие сервером соединения означает окончание результата. На выходе, мы, либо получим интересующую нас информацию, либо сообщение о ее отсутствии у данного whois-сервера.

Информация, которая приходит с whois-сервера, в случае успеха, выглядит как список объектов, каждый из которых содержит внутри себя набор данных, смотрите таблицу 1 [13]:

Таблица 1 – Список возвращаемых объектов whois-сервером

aut-num	Описание автономной системы
domain	DNS сервера
person / role	Административные и технические контакты лица / группы лиц
inetnum	Статус адресного пространства
route	Название автономной сети, которой принадлежит данная сети
as-set	Описывает (включает в себя) несколько автономных систем или других as-set.

Вот, к примеру, содержание объектов, перечисленных в таблице 1, из ответа одного whois-сервера:

– person:

person: Yaroslav V Kapsalov
address: JSC TransTeleCom
address: 7, Dolgorukovskaya st.
address: 127006 Moscow Russia
e-mail: y.kapsalov@transtk.ru
phone: +7 495 7846670
nic-hdl: YARK-RIPE
source: RIPE # Filtered

– inetnum:

inetnum: 217.150.32.0 - 217.150.32.255
netname: TTK-OFFICE-NET
descr: Transtelecom Office Network
descr: Moscow, Russia

country: RU
admin-c: KTTK-RIPE
tech-c: KTTK-RIPE
status: ASSIGNED PA
remarks: INFRA-AW
mnt-by: TRANSTELECOM-MNT
source: RIPE # Filtered

– domain:

domain: 32.150.217.in-addr.arpa
descr: Reverse delegation for TRANS-TELECOM-NET
remarks: INFRA AW
admin-c: KTTK-RIPE
tech-c: KTTK-RIPE
zone-c: KTTK-RIPE
nserver: dns-prim.transtk.ru
nserver: dns-sec.transtk.ru
nserver: ns.transtelecom.net
nserver: ns1.transtelecom.net
mnt-by: TRANSTELECOM-MNT
source: RIPE # Filtered

– route:

route: 217.150.32.0/19
descr: RU-TRANS-TELECOM-20010213
origin: AS20485
mnt-by: TRANSTELECOM-MNT
source: RIPE # Filtered

– aut-num:

aut-num: AS20485
as-name: TRANSTELECOM
descr: JSC Company TransTeleCom

descr: Moscow, Russia

– as-set:

as-set: AS-TTK
descr: Customers with TransTeleCom Global Access
members: AS20485
members: AS-CTTC
members: AS-KTTK
members: AS-SETTC
members: AS-SIBTTK
members: AS-STTK
members: AS-SUTTK
members: AS-TTKNN
members: AS-TTKNN-IZHEVSK
members: AS-UMN
members: AS-UMN-TMN
members: AS-VTT
members: AS-ZSTTK-SET
members: AS33989

Как можно увидеть, в объекте person, в полях адрес, содержится название страны, города и улицы.

Теперь нам необходимо взять необходимый IP-адрес и отправить запрос, содержащий данный IP, к его whois-серверу. Единственная проблема: как правильно определить whois-сервер для конкретного IP-адреса?

Есть пять основных региональных whois-серверов (см. рис. 1)[8]:

- whois.arin.net (Северная Америка)
- whois.apnic.net (Азия и Тихоокеанский регион)
- whois.ripe.net (Европа и Ближний Восток)
- whois.afrinic.net (Африка)
- whois.lacnic.net (Латинская Америка)

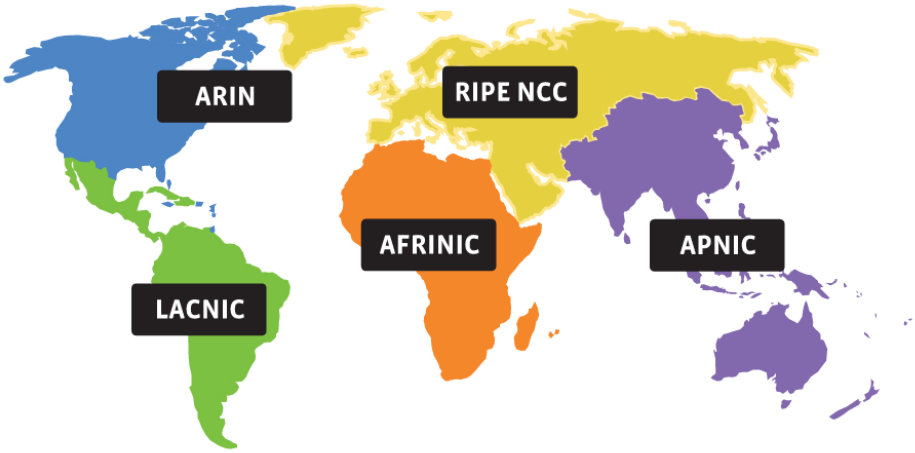

Рисунок 1 – Региональные whois-сервера

Теоретически, информация про любой IP адрес, должна быть на одном из них. Если же какой-то диапазон IP-адресов зарезервирован или ни одному из региональных серверов еще не выделен, то об этом должна знать IANA.

Итак, теперь мы знаем, что отправив запрос на iana.org, содержащий «неизвестный» IP-адрес, мы, либо получим в ответ интересующую нас информацию про данный адрес, либо название его whois-сервера, который, как правило, совпадает с региональным. В последнем случаем мы отправляем запрос на данный whois-сервер и получаем, опять же, либо нужные данные, либо имя ещё одного whois-сервера. Операцию повторяем, пока не найдем нужные данные или пока не получим сообщение об их отсутствии. Нет 100% гарантии получения информация, причины того: секретность некоторых адресов или отсутствие их владельцев, как таковых (адреса еще не делегированы).

2.3 Основная проблема при работе с whois-серверами

Алгоритм поиска информации из п. 2.2 (см. выше) хорошо работает в случае единичного IP-адреса, в нашем же случае, количество адресов может исчисляться тысячами. Проблема заключается в том, что большинство whois-серверов при получении большого количества запросов от одного адреса за короткий промежуток времени вводят штрафный санкции, в отношении источника этих запросов (прекращение отправки информации про IP-адресы). Этот факт делает невозможным проверку большого набора адресов с помощью whois-серверов.

Решением данной проблемы является использование базы данных IP-адресов. Ее наполнение и обновление происходит за счет тех же источников, что и наполнение whois-серверов. Одна их таких баз, предоставлена в открытом доступе, компанией MaxMind, занимающейся IP-аналитикой и сферой обнаружения интернет мошенничества[9]. Непосредственно в базе, в соответствие каждому IP-адресу сопоставлена информация в следующем виде:

country - страна

region - регион

city - город провайдера

ll - широта и долгота

Все параметры являются опциональными, т.е. часть из них или все (если IP-адреса в базе данных нет) могут отсутствовать.

2.4 Неизвестные адреса и traceroute

Следующий вопрос, который нас интересует – это, что делать с теми IP-адресами, информацию о которых мы не нашли, т.е. данные о них не содержаться в базе и поиск по whois-серверам ничего не дал. Решением данного вопроса стал инструмент traceroute, предназначенный для определения маршрутов следования данных в сетях.

Для определения промежуточных маршрутизаторов traceroute отправляет целевому узлу серию ICMP-пакетов (по умолчанию 3 пакета), с каждым шагом увеличивая значение поля TTL («время жизни») на 1. Это поле обычно указывает максимальное количество маршрутизаторов, которое может быть пройдено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и поэтому первый же маршрутизатор возвращает обратно ICMP-сообщение «time exceeded in transit», указывающее на невозможность доставки данных. Traceroute фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся на монитор компьютера). Затем traceroute повторяет отправку серии пакетов, но уже с TTL, равным 2, что заставляет первый маршрутизатор уменьшить TTL пакетов на единицу и направить их ко второму маршрутизатору. Второй маршрутизатор, получив пакеты с TTL=1, так же возвращает «time exceeded in transit».

Процесс повторяется до тех пор, пока пакет не достигнет целевого узла. При получении ответа от этого узла процесс трассировки считается завершённым. На оконечном хосте IP-датаграмма с TTL = 1 не отбрасывается и не вызывает ICMP-сообщения типа срок истёк, а должна быть отдана приложению. Достижение пункта назначения определяется следующим образом: отсылаемые traceroute датаграммы содержат UDP-пакет с заведомо неиспользуемым номером порта на адресуемом хосте. Таким образом, окончанием работы traceroute является ICMP-сообщением об ошибке «порт недоступен» [14]. В результате, в качестве ответа нам приходит список

адресов роутеров, вместе со временем ответа, через которые пришел наш запрос, прежде чем достиг заданного. Нам остается начать искать информацию привычным образом (с помощью whois и баз данных), по порядку в списке. Про первый из которых IP-адресов, мы найдем информацию, и будет нужный нам, а сама информация, будет соответствовать искомой.

2.5 Общий алгоритм

Обобщив все полученные данные, касательно поиска информации о IP-адресах, можно выстроить общий алгоритм, который реализован в нашем приложении. Алгоритм, который применяется последовательно к каждому IP-адресу в заданном списке, можно увидеть на рисунке 2.

Рисунок 2 – Схема общего алгоритма

Согласно алгоритму, в самом плохом случае (если применяем traceroute) , мы получим информацию, не об исходном IP-адресе, а о ближайшем его соседе. Мы применяем данный алгоритм ко всем адресам, из полученного списка с помощью traceroute, первый, про который мы найдем информацию, считается искомым. Таким образом, точность получения

данных уменьшается, но, тем не менее, информация полученная таким образом будет верна, с точность до региона.

3 Построение приложения

3.1 Общая концепция

Предполагается что система, должна уметь работать с большим количеством пользователей одновременно. К тому же, неплохим условием было бы то, чтобы система работала в режиме онлайн, с целью того чтобы мы смогли получить интересующую нас информацию из любой точки и в любое время. Вывод: нам нужно веб-приложение с клиент-серверной архитектурой, причем сервер должен уметь работать асинхронно. Помимо прочего, нам необходима утилита, позволяющая выводить информацию на географическую карту с возможностью динамического добавления и удаления объектов.

Приложение состоит из двух частей: сервер и контроллер (клиентская часть). Первый продуцирует и находит информацию, необходимую, для создания и финальной выдачи списка IP-адресов, вместе с информацией по каждому из них, а также выполняет некоторые другие функции, связанные с выдачей дополнительной информации, если на него придет, соответствующий запрос. Вторая, не менее важная часть – это контроллер, который отвечает за создание интерактивной формы (визуальной картины), путем отображения, полученной с сервера информации. Также, с помощью контроллера пользователь может сортировать информацию, для лучшего понимая общей картины.

Все составляющие написаны с использованием технологии JavaScript. Это асинхронный прототипно-ориентированный язык программирования. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование

обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания — что придаёт языку дополнительную гибкость [15]. Критериями выбора послужили такие факторы как асинхронность, автоматическая сборка мусора и динамическая типизация.

3.2 Серверная часть

Серверная часть приложения, отвечает за предварительную обработку списка IP-адресов, получение информации, касательно каждого адреса в списке, с помощью вышеприведенного алгоритма (п. 1.5) и отправку результата на контроллер.

Сервер было решено сделать средствами node.js. Это асинхронная платформа JavaScript, позволяющая создавать сетевые приложения и взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода [10].

MaxMind, для удобства использования их продуктов, предоставляют возможность подключить их geoip-библиотеку к нашему серверу, что существенно упрощает создание запросов к базе данных. Также используется библиотека под названием async.js. Это модуль позволяющий устанавливать последовательность действий, посредством callback-ов, в Javascript коде node.js сервера. Данный модуль способен работать с большинством известных объектов в Javascript [16]. Это особенно важно, учитывая асинхронность нашего сервера.

Сервер работает в режиме онлайн и способен работать с несколькими пользователями одновременно, отправляя необходимую информацию, по запросу. Содержит в себе, следующие модули (функции):

- Read – вычленение информации, из текстового файла, такой как список адресов, время их обращения и путь;
- GetInfo – получение информации о IP-адресах (содержит в себе infoFromWhois и GetRouteInfo);
- infoFromWhois – получение информации, посредством обращений к whois-серверам;

• GetRouteInfo – получение информации и вывод маршрута, посредством механизма traceroute;

• io.sockets.on – асинхронный модуль работы с пользователем, выполняет запросы пользователя в режиме онлайн (содержит в себе все вышеперечисленные модули, кроме Read).

Общий принцип работы следующий:

Сервер подготавливает список IP-адресов, предварительно взяв данные из текстового файла (с расширением txt), путь к которому прописан внутри сервера. Получает информацию про каждый IP-адрес в списке, в частности координаты или название города. Эту информацию, получая запрос от конкретного пользователя, сервер отправляет на контроллер. Далее он ждет, в случае если пользователь, запросит более подробную информацию, например исходный текст ответа от whois-сервера или полный маршрут следования данных (применить traceroute). В случае поступление, такого рода запроса, он его удовлетворяет или выводит сообщение о невозможности выполнения запроса.

3.3 Клиентская часть

Для реализации визуальной части была выбрана технология AngularJs. Это JavaScript платформа, разработанная компанией Google, специально для создания web-приложений, в частности одностраничных, с возможностью связи области ввода и обычных JavaScript переменных [17]. Одна из ключевых особенностей AngularJs, заключается в том, что он позволяет менять, в случае необходимости, лишь часть веб-страницы, полностью не обновляя её, это особенно важно т.к. количество записей, с которыми нам придется работать, исчисляется тысячами. Поэтому, перезагрузка всей страницы будет нам дорого стоить, в смысле памяти и времени отклика страницы. AngularJs позволяет нам этого избежать, с помощью своей системы контроллеров и областей видимости.

Также, нам необходим инструмент, позволяющий вывести карту с возможностью рисования на ней различных объектов - реализация картодиаграммы. Данным инструментом явился для нас Google Maps. Критериями выбора явились [18]:

- легко подключаемая API;
- бесплатный доступ;
- возможность установки объектов на карту, как стандартных, так и собственного производства (это особенно важно при представлении источников трафика);
- легко синтезируется с другими JavaScript платформами.

Таким образом, клиентская часть приложения (сама web-страница), включает в себя 2 основные технологии: AngularJs и Google Maps. За все изменения отвечает контроллер, в нем же, содержаться все необходимые функции к приему с сервера и вывода на страницу данных:

- initialize – функция первичной инициализации страницы, делает запрос на сервер с целью, получения интересующего нас набора IP-адресов и информацию к ним;
- moreinfo – функция вызывается, если пользователь запросил более полную информацию касательно конкретного ip-адреса (ответ whois-сервера, маршрут);
- socket.on – комплекс функции, призванных принимать, обрабатывать и корректно отображать данные на страницы, с сервера.

Расставление источников на карте происходит с помощью создания объектов типа «маркер». Данный объект выставляется на карту согласно координатам источника, если информация на сервере была получена из базы данных, или согласно городу и стране (Google Maps умеет самостоятельно выдавать координаты по названию страны и города), если информация на сервере была получена из ответа whois-сервера.

Как было сказано выше, количество объектов, изображающих источники интернет-трафика, может исчисляться тысячами, в связи с этим может происходить наложение множества объектов друг на друга и загромождение, что как следствие является ухудшением восприятия картины. Результатом решения данной проблемы, является реализация структурных диаграмм, с помощью библиотеки Google maps clusters [19]. Она позволяет представить множество источников (в нашем случае маркеров), как совокупность структурных диаграмм, с числом соответствующем числу источников в данной области, можно увидеть на рисунке 3.

Рисунок 3 - Структурные диаграммы

Которые распадаются на более мелкие объекты того же типа и источники (маркеры), что можно увидеть на рисунке 4.

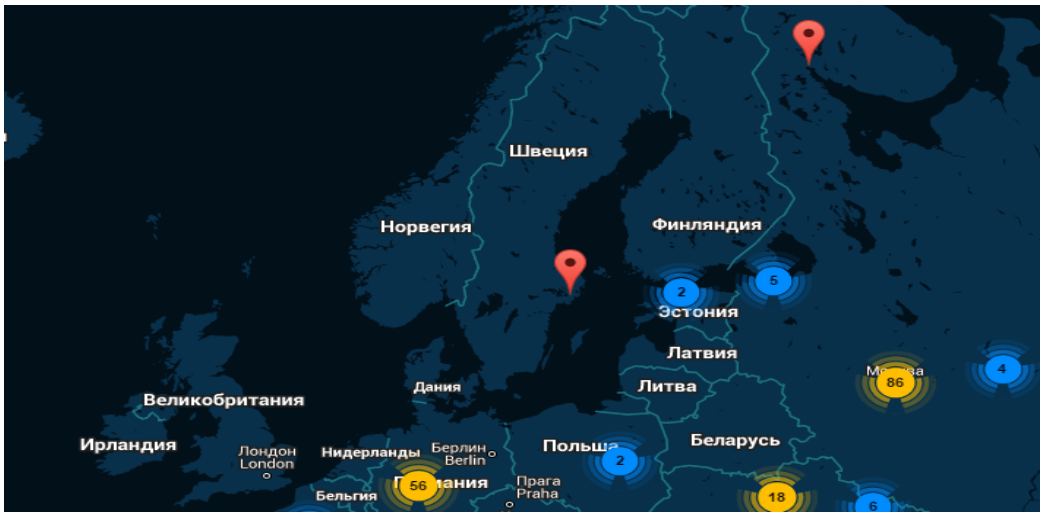

Рисунок 4 – При приближении

Поступая на контроллер, информация проходит предварительную обработку, и выводится, путем рисования маркеров в соответствии с их положением (по широте и долготе), пример работы представлен на рисунке 5.

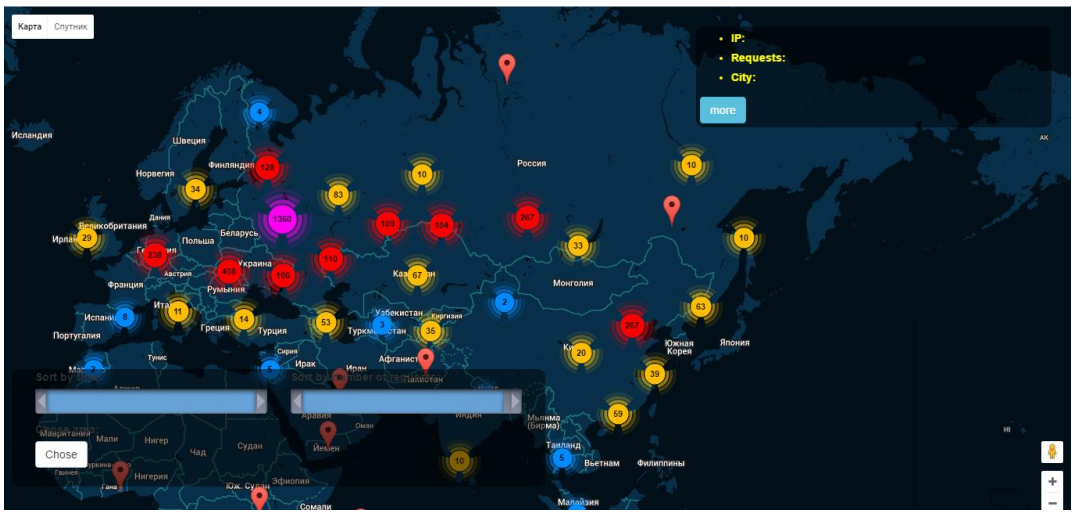

Рисунок 5 – Пример работы приложения в целом

Помимо всего прочего, пользователь может получить расширенную информацию относительно интересующего его IP-адреса. Нажав кнопку «more», пользователь получит исходный текст ответа от whois-сервера, маршрут до соответствующего адреса в виде красной линии с анимированной стрелкой и список IP-адресов роутеров, вместе с их координатами и отображением на карте, через которые этот маршрут проходит. Это можно увидеть на рисунке 6.

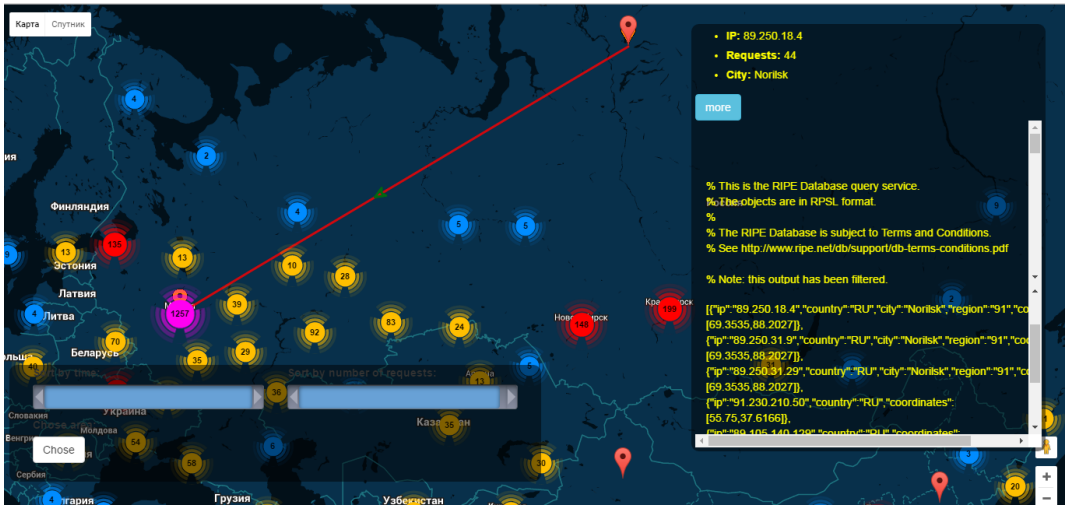

Рисунок 6 – Получение подробной информации про IP-адрес

Также существует возможность, видеть IP-адреса только с интересующих нас регионов, с возможностью сортировать по количеству запросов и времени обращения к IP-адресу, смотрите рисунок 7.

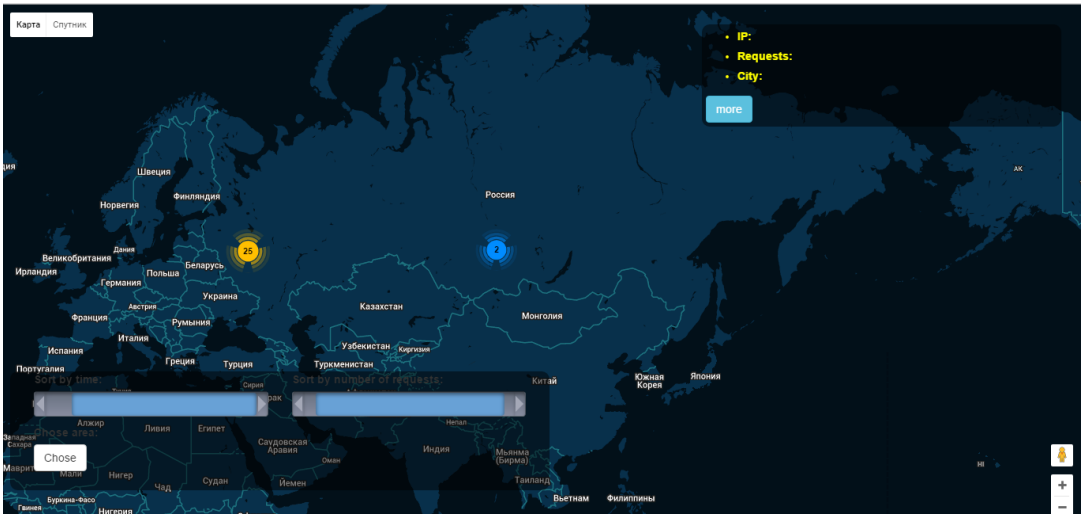

Рисунок 7 – Выбрана только Россия и количество запросов

Больше 170

3.4 Работа системы

Структура приложения выглядит следующим образом:

Server.js – основной сервер, используемые технологии (библиотеки):

- Node.js;
- Socket.io;
- Geoip;
- Async.js.

MainController.js + index.html – Контроллер и связанная с ним страница, используемые технологии (библиотеки):

- Google Maps;
- AngularJs;
- Google Maps clusters;
- jquerySlider;
- Socket.io.

Мы построили клиент-серверное приложение, способное работать в режиме реального времени и с большим количеством пользователей. Данное приложение находит информацию касательно интересующего нас списка IP-адресов на сервере (server.js), после отправляет ее на контроллер. Данный список получается, путем «парсинга» текстового файла, путь к которому прописан к серверу, поэтому для просмотра визуализации другого журнала интернет-трафика, необходимо прописать путь к нему непосредственно к серверу (или копировать, с заменой, содержимое интересующего журнала в текущий).

Контроллер выводит полученные данные на географическую карту, предварительно проведя их кластеризацию. Карта располагается на веб-странице, где пользователь может взаимодействовать с ней, масштабируя ее, а также получать более подробную информацию относительно интересующего его IP-адреса (такую как маршрут следования данных и

исходный ответ от whois-сервера). Также существует возможность, рассматривать источники лишь в заданных регионах или в заданных промежутке времени.

ЗАКЛЮЧЕНИЕ

В бакалаврской работе проведено исследование визуализации интернет-трафика. Рассмотрены соответствующие сервисы. Обоснована важность решения задачи визуализации данных об источниках интернет-трафика. Рассмотрены методы и способы получения информации об источниках. Построен соответствующий алгоритм поиска данных относительно IP-адресов. На основе этого алгоритма, сделано клиент-серверное приложение, способное работать в режиме реального времени, предоставляя пользователям визуальную картину в виде картодиаграммы источников интернет-трафика. Данное приложение построено с помощью наиболее удобных и современных технологий, таких как: node.js, AngularJs и Google maps.

Приложение может быть использовано для визуализации источников интернет-трафика, что позволяет географически классифицировать интерес злоумышленников к вашему веб-ресурсу (серверу, сайту). Или наоборот, поможет выяснить географию исходящих от вас запросов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Организация Wireshark [Электронный ресурс] – режим доступа: https://www.wireshark.org/

2 Маркин, Ю.В. Обзор современных инструментов анализа сетевого трафика [Электронный ресурс] – режим доступа: http://www.ispras.ru/preprints/docs/prep_27_2014.pdf

3 Организация Google [Электронный ресурс] – утилита Google Analytics – режим доступа:

https://www.google.com/intl/ru_ALL/analytics/features/analysis-tools.html

4 Герчук, Я. П. Графические методы в статистике: учебное пособие / Я.П. Герчук. – Москва : Статистика, 1987.

5 Григорьев, А. А. Краткая географическая энциклопедия [Электронный ресурс] : параграф 3.11 / А. А. Григорьев – 2001. – Режим доступа: http://geoman.ru/books/item/f00/s00/z0000060/st028.shtml

6 Крейг Х. Персональные компьютеры в сетях TCP/IP: Пер. с англ. — К.: Издательская группа BHV, 1997 — 384 с

7 Организация IANA [Электронный ресурс] – режим доступа: http://www.iana.org

8 Организация IANA [Электронный ресурс] – о делегировании адресного пространства – режим доступа: http://www.iana.org/numbers

9 Организация MaxMind [Электронный ресурс] – раздел о компании – режим доступа: https://www.maxmind.com/ru/about-maxmind

10 Организация Nodejs [Электронный ресурс] – режим доступа: https://nodejs.org/en/about/

11 Mockapetris, P. Request for Comments 1035 / P. Mockapetris - The Internet Engineering Task Force, – 1987

12 Daigle, L. Request for Comments 3912 / L. Daigle - The Internet Engineering Task Force, – 2004

13 RIPE network coordination center [online source] – list of primary objects – access mode: https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation/ripe-database-structure/3-1-list-of-primary-objects

14 Malkin, G. Request for Comments 1393 / G. Malkin - The Internet Engineering Task Force, – 1993

15 Kowan, K. CommonJS effort sets JavaScript on path for world domination – arstechnica / K. Kowan - 2009

16 Early, A. Documentation for async.js [online source] – documentation – access mode: https://github.com/caolan/async

17 Chris, S. Angular Basics [online source] – access mode: http://www.angularjsbook.com/angular-basics/chapters/introduction/

18 Google Developers [online source] – documentation – access mode: https://developers.google.com/maps/documentation/javascript/

19 Google Developers [online source] – google maps clustering - access mode: https://developers.google.com/maps/articles/toomanymarkers